

Support d'apprentissage/ de formation

Siemens Automation Cooperates with Education (SCE) | À partir de NX MCD V12/TIA Portal V15.0

Module DigitalTwin@Education 150-006 Création de signaux pour un modèle 3D dynamique dans le système d'IAO Mechatronics Concept Designer

siemens.com/sce

SIEMENS

Global Industry Partner of WorldSkills International



Diffusion non restreinte pour les instituts publics de formation et de R&D. © Siemens 2020. Tous droits réservés.

Packs pour formateurs SCE correspondant à ce support d'apprentissage/de formation

SIMATIC STEP 7 Software for Training (y compris PLCSIM Advanced)

- SIMATIC STEP 7 Professional V15 Licence monoposte N° de référence : 6ES7822-1AA05-4YA5
- SIMATIC STEP 7 Professional V15 Licence pour salle de classe 6 postes N° de référence : 6ES7822-1BA05-4YA5
- SIMATIC STEP 7 Professional V15 Licence de mise à niveau 6 postes N° de référence : 6ES7822-1AA05-4YE5
- SIMATIC STEP 7 Professional V15 Licence 20 postes étudiants N° de référence : 6ES7822-1AC05-4YA5

Logiciel SIMATIC WinCC Engineering/Runtime Advanced dans TIA Portal

- SIMATIC WinCC Advanced V15 Licence pour salle de classe 6 postes 6AV2102-0AA05-0AS5
- Mise à niveau SIMATIC WinCC Advanced V15 Licence pour salle de classe 6 postes 6AV2102-4AA05-0AS5
- SIMATIC WinCC Advanced V15 Licence 20 postes étudiants 6AV2102-0AA05-0AS7

NX V12.0 Educational Bundle (écoles, universités, pas pour les établissements de formation professionnels)

Interlocuteur : <u>academics.plm@siemens.com</u>

Plus d'informations sur le programme SCE

siemens.com/sce

Remarque d'utilisation

Le support d'apprentissage/de formation SCE pour une solution d'automatisation Totally Integrated Automation (TIA) cohérente a été créé spécialement pour le programme **"Siemens Automation Cooperates with Education (SCE)"** à des fins de formation pour les instituts publics de formation et de R&D. Siemens n'assume aucune responsabilité quant au contenu.

Cette documentation ne peut être utilisée que pour une première formation aux produits/systèmes Siemens. Autrement dit, elle peut être copiée, en partie ou en intégralité, pour être distribuée aux étudiants/participants à la formation afin qu'ils puissent l'utiliser dans le cadre de leur formation. La diffusion et la duplication de cette documentation, l'exploitation et la communication de son contenu sont autorisées au sein d'instituts publics de formation et de formation continue ou dans le cadre de la formation.

Toute exception requiert au préalable l'autorisation écrite de la part de Siemens. Envoyer toutes les demandes à ce sujet à <u>scesupportfinder.i-ia@siemens.com</u>.

Toute violation de cette règle expose son auteur au versement de dommages et intérêts. Tous droits réservés, également pour la traduction, en particulier en cas de délivrance de brevet ou d'enregistrement d'un modèle déposé.

Il est expressément interdit d'utiliser cette documentation pour des cours dispensés à des clients industriels. Tout usage de cette documentation à des fins commerciales est interdit.

Nous remercions l'université de Darmstadt, en particulier Mr Heiko Webert, M.Sc. et Mr le Professeur Dr.-Ing. Stephan Simons, ainsi que toutes les personnes ayant contribué à la réalisation de ce support d'apprentissage/de formation.

Sommaire

1		Obje	ectif	7
2		Con	ditions	7
3		Con	figurations matérielles et logicielles requises	8
4		Thé	orie	9
	4.	1	Communication avec des sources externes	9
	4.	2	Propriétés de signaux dans Mechatronics Concept Designer	10
5		Éno	ncé du problème	12
6		Plar	ification	12
7		Instr	ructions structurées par étapes	13
	7.	1	Création des signaux pour le modèle dynamique	14
	7.	2	Création d'une connexion de signal entre un API virtuel et un jumeau numérique	30
	7.	3	Test du jumeau numérique avec l'API virtuel	36
8		Liste	e de contrôle – Instructions structurées par étapes	37
9		Info	rmations supplémentaires	38

Liste des figures

Figure 1 : Vue d'ensemble des composants logiciels et matériels requis dans ce module	8
Figure 2 : Application "Mechatronics Concept Designer" dans NX avec marquages pour expliquer I zones	es 10
Figure 3 : Recherche de commande dans le menu NX, marquée en orange	13
Figure 4 : Ouverture d'un module dans NX	14
Figure 5 : Ajout de paramètres de propriétés dynamiques pour des signaux dans l'adaptateur de signa	ux 15
Figure 6 : Propriété Read/Write (Lecture/écriture) d'un paramètre	16
Figure 7 : Création d'un signal approprié pour un paramètre	17
Figure 8 : Définition d'une formule entre signal et paramètre	18
Figure 9 : Création d'un signal de sortie pour un capteur photoélectrique	19
Figure 10 : Formule pour le signal du système de capteurs photoélectriques "csLightSensorCylinder"	20
Figure 11 : Création d'un signal de vitesse de type de données "double"	23
Figure 12 : Création d'un paramètre pour une surface de transport	25
Figure 13 : Création de l'adaptateur de signaux "saSortingPlant"	26
Figure 14 : Initier la création d'une nouvelle table des mnémoniques pour l'adaptateur de signaux	27
Figure 15 : Terminer la création d'une nouvelle table des mnémoniques pour l'adaptateur de signaux?	28
Figure 16 : Terminer l'affectation de mnémoniques pour l'adaptateur de signaux	29
Figure 17 : Sélection d'une affectation de signal via PLCSIM Advanced	31
Figure 18 : Valider des variables de l'instance PLCSIM Advanced pour l'affectation de signaux	32
Figure 19 : Affectation d'un signal MCD à un signal externe	33
Figure 20 : Connecter tous les signaux par mappage automatique	34
Figure 21 : Couper le mappage du signal	35
Figure 22 : Confirmation du signal entre modèle dynamique et API virtuel	35

Liste des tableaux

Création de signaux pour un modèle 3D dynamique dans le système d'IAO Mechatronics Concept Designer

1 Objectif

Dans le Module 4 du cycle de formation DigitalTwin@Education, vous avez créé de manière autonome le modèle 3D statique d'une installation de tri. Le résultat est un module dans lequel les composants individuels requis de l'installation de tri ont été insérés et positionnés correctement dans l'espace. Sur cette base, le Module 5 met l'accent sur la dynamisation du modèle 3D. Grâce à l'affectation des propriétés physiques, les composants de l'installation de tri peuvent interagir les uns avec les autres.

Pour que le jumeau numérique interagisse avec un API virtuel, vous avez enfin besoin d'une connexion entre Mechatronics Concept Designer (MCD) et PLCSIM Advanced qui sera utilisée pour simuler l'API avec le programme d'automatisation. Le but de ce module est de créer des signaux et de les mapper sur les deux programmes. Vous utiliserez ensuite le programme d'automatisation du Module 1 du cycle de formation DigitalTwin@Education pour valider le fonctionnement correct de votre jumeau numérique.

2 Conditions

Des connaissances sur les propriétés dynamiques du modèle utilisées dans le Module 5 sont requises pour ce module. En outre, vous devez comprendre le fonctionnement du programme d'automatisation présenté dans les Modules 1 - 2, car vous utiliserez de nouveau le programme dans ce module.

3 Configurations matérielles et logicielles requises

Les composants suivants sont requis pour ce module :

- 1 Station d'ingénierie : Les prérequis sont le matériel et le système d'exploitation (pour plus d'informations : voir le fichier Readme/Lisezmoi sur les DVD d'installation de TIA Portal et dans le pack logiciel NX)
- 2 Logiciel SIMATIC STEP 7 Professional dans TIA Portal à partir de V15.0
- 3 Logiciel SIMATIC WinCC Runtime Advanced dans TIA Portal à partir de V15.0
- 4 Logiciel SIMATIC S7-PLCSIM Advanced à partir de V2.0
- 5 Logiciel NX avec extension Mechatronics Concept Designer à partir de V12.0



Figure 1 : Vue d'ensemble des composants logiciels et matériels requis dans ce module

La <u>Figure 1</u> montre que la station d'ingénierie est le seul composant matériel du système. Tous les autres composants sont exclusivement logiciels.

4 Théorie

4.1 Communication avec des sources externes

Dans le Module 5 de ce cycle de formation, vous avez créé des propriétés dynamiques et avez testé leur fonctionnalité à l'aide du Runtime Inspector (Surveillance de l'exécution) dans Mechatronics Concept Designer. Pour un jumeau numérique, cependant, il est approprié d'établir une connexion à un API, afin que ce dernier puisse modifier des propriétés dynamiques dans MCD et que les résultats de MCD soient mis à la disposition de l'automate.

MCD fournit un grand nombre de manières de communiquer avec des programmes externes (voir <u>Chapitre 9</u>, lien [1]). Il s'agit entre autres de :

- la communication avec MATLAB via le protocole MATLAB
- l'établissement d'une liaison à un serveur OPC (à partir de MCD V12.01 également à un serveur OPC UA)
- la communication S7 via PLCSIM Advanced ou directement via le protocole PROFINET
- la liaison à une mémoire partagée (Shared Memory), par ex. pour SIMIT
- les liaisons TCP/UDP

Dans ce module, vous configurerez la communication avec PLCSIM Advanced, comme dans les Modules 1 - 3 de ce cycle de formation. Cela permet à un API virtuel de se connecter à NX/MCD.

4.2 Propriétés de signaux dans Mechatronics Concept Designer

La communication avec des programmes externes est configurée dans l'application NX Mechatronics Concept Designer par la définition et le mappage de signaux.

La surface de travail de Mechatronics Concept Designer est présentée à la <u>Figure 2</u>. Pour ouvrir cette application dans NX, vous pouvez utiliser la recherche de commande en haut à droite de l'écran pour rechercher l'application "**Mechatronics Concept Designer**".



Figure 2 : Application "Mechatronics Concept Designer" dans NX avec marquages pour expliquer les zones.

Les fenêtres suivantes sont utilisées dans cette application pour la définition de signaux et pour le test du jumeau numérique :

- L'écran central (voir <u>Figure 2</u>, zone 1) comprend la surface de travail tridimensionnelle dans laquelle vous pouvez suivre le fonctionnement de votre modèle 3D dynamique en association avec l'API virtuel pendant une simulation.
- Vous pouvez commander la simulation de votre modèle dans la partie centrale de la barre de menu (voir <u>Figure 2</u>, zone 2) Vous utiliserez ces fonctions au <u>Chapitre 7.3</u>.

- Vous trouverez les propriétés de signaux dans la zone des propriétés électriques de la barre de menu, directement à côté des propriétés mécaniques dynamiques (voir Figure 2, zone 3). Vous pouvez y créer de signaux et des tableaux. Certaines commandes sont expliquées brièvement dans les paragraphes suivants.
 - Vous pouvez utiliser la commande **Signal** pour créer dans votre modèle un 0 signal permettant de contrôler des propriétés physiques d'un objet à l'aide d'une expression d'exécution. Une expression d'exécution est une valeur non statique pouvant changer pendant l'exécution d'une simulation. Cette expression est connectée en interne à MCD où elle est déterminée par une connexion à un signal d'une source externe comme PLCSIM Advanced.
 - En créant une table des mnémoniques vous définissez une liste de 0 mnémoniques utilisés pour la désignation univoque de signaux. Vous pouvez également importer une table des mnémoniques à partir d'une source externe, comme par ex. STEP 7.
 - Des signaux et des expressions d'exécution peuvent être interconnectés à l'aide de 0
 - ~~L

l'adaptateur de signaux . Il est possible d'utiliser plusieurs signaux et expressions d'exécution pour chaque adaptateur de signaux. En outre, cette commande permet de créer des signaux et des expressions d'exécution.

- La barre de menu de MCD contient également les propriétés de signaux pour l'automatisation (voir Figure 2, zone 4). Ici, vous utiliserez la propriété suivante :
 - Le Signal Mapping (Affectation de signaux) interconnecte des signaux de MCD avec des signaux de programmes externes. Il peut s'agit par ex. de PLCSIM Advanced.
- À partir de la barre de ressources à gauche de l'écran dans MCD (voir Figure 2, zone 5), vous pouvez entre autres ouvrir le Physics Navigator (navigateur physique). Vous y stockerez vos signaux et connexions.



Pour plus d'informations sur les propriétés de signaux dans Mechatronics Concept Designer, vous pouvez rechercher les entrées correspondantes dans l'aide en ligne (voir le Chapitre 9, lien [2]).

Pour effectuer la recherche, il est cependant recommandé d'utiliser des termes anglais car l'aide en ligne ne comprend qu'un ensemble limité de termes français.

Diffusion non restreinte pour les instituts publics de formation et de R&D. © Siemens 2020. Tous droits réservés. sce-150-006-mcd-tia-com-digital-twin-at-education-signal-mapping-mcd-hs-darmstadt-0720-fr.docx

5 Énoncé du problème

Dans l'étape suivante, vous devez ajouter des signaux au modèle 3D dynamique de l'installation de tri que vous avez créée dans le Module 5 et établir une connexion à un API virtuel. De plus, vous devez charger le programme d'automatisation du Module 1 du cycle de formation DigitalTwin@Education dans l'API virtuel et valider votre propre jumeau numérique.

Vous aurez à nouveau besoin de l'application NX Mechatronics Concept Designer (MCD) NX pour cela. Cette fois, cependant, vous devez vous concentrer uniquement sur la connexion de votre modèle 3D dynamique à des programmes externes.

6 Planification

Le système de CAO **NX** en version **V12.0** ou ultérieure est requis pour l'affectation de signaux à un modèle 3D dynamique et la mise en service de ce modèle. Le module complémentaire **Mechatronics Concept Designer (MCD)** doit également être disponible dans NX.

Vous devez avoir des connaissances sur les **modèles 3D statiques** et **dynamiques**, que vous avez acquises dans les Modules 4 et 5. Vous devez également rafraîchir vos connaissances sur le **fonctionnement du programme d'automatisation** présenté dans les **Modules 1 – 2** du cycle de formation DigitalTwin@Education. En cas de doute sur le fonctionnement de l'installation de tri, consultez en particulier la section Théorie du **Chapitre 4.2** du **Module 1**.

Familiarisez-vous de nouveau avec l'interaction entre l'API virtuel et le jumeau numérique du **Module 1** et gardez la description du Module 1 de ce cycle de formation à portée de main car vous en aurez besoin, en particulier aux <u>Chapitres 7.2</u> et <u>7.3</u>.

Le guide de normalisation "**Guide to Standardization**" de Siemens a été pris en compte pour la dénomination des signaux. Vous trouverez ce guide sous le lien [3] au <u>Chapitre 9</u>.

7 Instructions structurées par étapes

Le projet "**150-006_DigitalTwinAtEducation_NX_dynModelSignals**" est fourni dans ce module. Le dossier comprend trois sous-dossiers :

- "fullDynModel" contient le modèle 3D dynamique complet de l'installation de tri du Module 5. Vous pouvez utiliser ce modèle comme base pour ce module si vos résultats du Module 5 sont incomplets.
- "fullDigTwin" contient le jumeau numérique complet et donc la solution pour ce module. Il sert d'assistance si vous avez besoin d'aide pour compléter une étape.
- "fullPlcBasic" fournit le programme d'automatisation avec IHM intégrée que vous connaissez du Module 1. Vous en aurez besoin pour tester votre jumeau numérique.

Pour rappel, utilisez la recherche de commande si vous ne trouvez pas une commande ou une application dans l'environnement de développement dans ce module. Elle est située dans la partie supérieure droite de l'écran d'interface utilisateur de NX, comme indiqué à la Figure 3.

NX) 🗟 • 🛷	Switch W	indow 🜅	Window	/ • •		NX 12		_		×
Fil	e Ho	ome Tool	s 3Dconne	xion					Find a Command 🔎		\diamond	0
		2		S	a.		?					
Ne	w Open	Open a Recent Part •	Assembly Load Options	Customer Defaults	Touch Mode	Window	Help •					•
(1)	Menu 🕶											•
Ø	History				🐎 Welc	ome Page	×					

Figure 3 : Recherche de commande dans le menu NX, marquée en orange

Vous pouvez choisir la commande appropriée dans les occurrences trouvées. NX vous montre également où la commande se situe pour vous permettre de la sélectionner ultérieurement directement depuis le menu.

<u>Important</u> : l'interface utilisateur et la disposition de différentes commandes dans les menus ont été modifiées dans les nouvelles versions de NX. De plus, les utilisateurs peuvent définir leur propre interface utilisateur. Les descriptions suivantes se réfèrent à l'interface utilisateur standard de NX12.0 ; l'interface utilisateur peut être différente dans votre version. **Utilisez par conséquent la recherche de commande si vous ne trouvez pas une commande à la position décrite dans la fenêtre.**

Tenez également compte du fait que cette description n'est qu'une proposition de solution. Le but est ici de décrire une procédure facile à comprendre permettant à votre jumeau numérique d'interagir simplement avec une CPU virtuelle des Modules 1-3.

Notez que certains passages de ce module sont mis en évidence sous forme de sections. Comme il est souvent fait référence à ces passages dans cette description, ces marquages servent d'orientation.

7.1 Création des signaux pour le modèle dynamique

Dans ce chapitre, vous devez créer pour votre installation de tri tous les signaux requis qui doivent être commandés en externe par un API. Pour ce faire, procédez comme suit :

- → Créez une copie des modèles créés au Module 5 avec votre système d'exploitation. Enregistrez-la dans un nouveau dossier dans votre système de fichiers. Comme mentionné au <u>Chapitre 7</u>, vous pouvez également accéder au projet "fullDynModel" fourni et créer une copie de travail à partir de ce dossier si votre modèle dynamique est incomplet.
- → Démarrez NX et attendez que le programme s'ouvre et que la page d'accueil soit affichée. Cliquez sur le bouton "Open" (Ouvrir) (voir Figure 4, étape 1) et naviguez jusqu'au dossier que vous avez créé auparavant. Vous voyez maintenant les pièces créées dans le Module 5. Sélectionnez le module "assSortingPlant" qui contient le modèle 3D dynamique complet de l'installation de tri (voir Figure 4, étape 2). Sélectionnez l'option "Partially Load" (Partiellement chargé) (voir Figure 4, étape 3) pour que seuls les modèles et les propriétés dynamiques des composants individuels du modèle soient chargés, et pas les dessins ou les systèmes de coordonnées additionnels. Confirmez enfin votre en cliquant sur "OK" (voir Figure 4, étape 4).

NX C) 🤔 🤌 - 🥔 📴 s	witch Window 📘 Window 🕶 🗟	Θ	
File	Home Tools 3	Dconnexion	(1) (2)	
	3	🕫 📲 📣 📃 🕻	2	
New O	Re 🔮 Open			×
🚡 Menu	Suchen in:	SortingPlant_dynModel	- ← 🗈 👉 💷 -	
O Hist	tony 🛃	Name	Änderungsdatum	Тур
~ Hist	Schoollzundff	SassSortingPlant	12.02.2020 16:53	Siem
	Scrineizügni	Sontainer 🔊	17.10.2019 11:59	Siem
		Si conveyorLong	15.10.2019 12:39	Siem
	Desktop	SonveyorShort	15.10.2019 12:24	Siem
2		2 cylinderHead	17.10.2019 15:55	Siem
Ou		Second Se	21.10.2019 14:08	Siem
	Bibliotheken	2 lightRay	21.10.2019 15:41	Siem
		2 lightSensor	21.10.2019 15:41	Siem
	Dieser PC	2 lightSensor_mirror	20.01.2020 16:07	Siem
	Dicserre	2 limitSwitchSensor	17.01.2020 13:30	Siemi IV Preview
		workpieceCube	14.10.2019 12:46	Siem
-	Netzwerk	workpieceCylinder	14.10.2019 16:25	Siem
3.				
		<		<u>,</u> (4)
1		Dateiname: assSortingPlant	•	ок
	3	Dateityp: Part Files (*.prt)	Abbr	rechen
	Load Structure	Only		
	Option Parti	ally Load		
		-,		
	Options			1

Figure 4 : Ouverture d'un module dans NX

Section : Création et connexion de signaux avec l'adaptateur de signaux

→ Une fois le module ouvert dans l'application NX "Mechatronics Concept Designer", vous créez votre premier exemple. Pour cela, vous devez créer et connecter un signal permettant d'activer la source d'objet, requis pour générer des pièces à usiner parallélépipédiques. Pour ajouter des signaux et interconnecter leurs propriétés dynamiques, commencez par ouvrir la commande "Signal Adapter" (Adaptateur de signaux) dans la zone "Electrical" (Électrique) comme représenté à la Figure 5, étape 1. La fenêtre de commande "Signal Adapter" (Adaptateur de signaux) s'ouvre. Sélectionnez tout d'abord un paramètre d'une propriété dynamique à connecter à un signal. Pour cela, cliquez sur le bouton "Select Physics Object" (Sélectionner l'objet physique) dans la zone de commande "Parameters" (voir Figure 5, étape 2). Naviguez dans la barre de ressources jusqu'au "Physics Navigator" (Navigateur physique) (voir Figure 5, étape 3) et sélectionnez-y la source d'objet "osWorkpiece Cube" comme premier paramètre (voir Figure 5, étape 4). Une fois cette sélection effectuée, vous pouvez sélectionner dans la fenêtre de commande le paramètre correspondant que vous voulez affecter à un signal de la liste "Parameter Name" (Nom de paramètre). Dans ce cas, choisissez le nom de paramètre "active" (actif) de la source d'objet sélectionnée (voir Figure 5, étape 5) Cliquez sur le bouton "Add Parameter" (Ajouter un paramètre) (voir Figure 5, étape 6) pour ajouter le paramètre dans cet adaptateur de signaux.

File Req Syster	e Home Modeling Assemblies	Curve Ar ay (4) II 2 ⁽²⁾ op C 2 ulate	nalysis Viev \vec{o} · \vec{o} \vec{o} · \vec{o} \vec{o} · \vec{o} \vec{o} · \vec{o}	v Render Angular Spring Linear Spring J Angular Limit J Mechanical	Tools Applie Joint 1	ation 3Dcor	Automation	d a Command Q Add Design Colla.	
T	Menu - No Selection Filter - Entire A	sembly 🔻	±3 ∾, 4,	• 🍾 🦏 6	🌡 🗔 🔹 🦚 🕻) 🗖 🗐	0 🧃 🖢	🔣 • 🕹 • 🏐	- 🐼 -
ø	Physics Navigator	assSo	rtingPlant.prt	×					
	Name 🔺	🗘 Sigr	al Adapter						٢
3_	😑 🚘 Basic Physics	Param	eters		6				
						2			4
-0-	• osWorkpieceCylinder	V Sel	ect Physics Ob	ject (1)					Ψ
	+ M @ rbContainer	Parame	ter Name		5		a	ctive	•
Ø"	+ M G rbConveyorLong	Add Pa	rameter		C				*
	+ M G rbCvlinderHead	AA	ias	Object	Object Ty P	arameter V	alue	Jnit Ja	ta Tyr 🗙
P1= P2=	rbCylinderLiner							(6)	
a.								U	
Fa	+ 🗹 🥳 rbWorkpieceCylinder	<							> +
M	- 🔁 Joints and Constraints	Signals							1
	- ₩ FjContainer							10.00	
a_		AN	ame	Data lype	Input/Out	Initial Value	Measure	Unit	+
Fo	- ₩ fjConveyorShort	×							×
<u> </u>	< >>	-							\$
÷	Details	·					OK	Annel	
100	Dependencies						OK	Apply	Cance

Figure 5 : Ajout de paramètres de propriétés dynamiques pour des signaux dans l'adaptateur de signaux

→ Le paramètre que vous venez de sélectionner est maintenant visible dans le tableau dans la zone "Parameters" (Paramètres). Modifiez son alias en "paOsWorkpieceCube_SetActive" (voir Figure 7, étape 1). Le préfixe "pa" signifie "parameter" (paramètre) afin de faire une distinction claire avec un nom de signal. Cliquez également dans la case au début du tableau pour affecter ultérieurement un signal au paramètre. Cela est indiqué par une coche M. Si vous faites défiler vers la droite la zone "Parameters" (Paramètres), vous pouvez voir d'autres propriétés du paramètre, notamment la propriété "Read/Write" (Lecture/écriture) qui indique si un paramètre est accessible en lecture ("R" pour Read/lecture) ou en écriture ("W" pour Write/écriture). Le paramètre actuel "paOsWorkpieceCube_SetActive" est uniquement accessible en écriture (voir Figure 6, étape 1).

Signal Adapter						ა x
Parameters						^ ^
★ Select Physics Object (0)				\bigcirc		+
Parameter Name				$\left(1\right)$		Ŧ
Add Parameter						*
A Alias	Object	Р	V.	Data Type	Read/Write	X
paOsWorkpieceCube_SetActive	osWorkpieceCube	a.	t	bool	W	(

Figure 6 : Propriété Read/Write (Lecture/écriture) d'un paramètre

→ Vous avez maintenant besoin d'un signal correspondant auquel le paramètre doit être connecté. Pour cela, cliquez sur le bouton "Add" (Ajouter) dans la zone de commande "Signals" (Signaux) (voir Figure 7, étape 2). Un nouveau signal apparaît. Adaptez les propriétés du signal en fonction du paramètre. Pour cela, double-cliquez sur le nom par défaut du signal "Signal_0" et renommez-le en "osWorkpieceCube_SetActive". Le même type de données doit être sélectionné pour le signal et pour le paramètre associé. Dans ce cas, "bool". La valeur de la propriété "Input/Output" (Entrée/sortie) doit être sélectionnée en fonction de la propriété "Read/Write" (Lecture/écriture) du paramètre. Pour un paramètre accessible en écriture, le signal doit être une entrée provenant d'une source externe du point de vue de MCD. Pour un paramètre accessible en lecture, le signal doit être une sortie vers un programme externe. Comme le paramètre actuel "paOsWorkpieceCube_SetActive" est accessible en écriture, la valeur "Input" (Entrée) doit être sélectionnée pour le signal "osWorkpieceCube_SetActive". La valeur initiale sélectionnée dans le tableau doit être équivalente à la valeur initiale du signal. Dans ce cas, elle doit être "false" (voir Figure 7, étape 3).

Support d'apprentissage/de formation | Module DigitalTwin@Education 150-006 | Édition 07/2020 | Digital Industries, FA

NX	🖬 🤊 • (° 🖌 🗅 🛱 🚯	7 <mark>2</mark> -	🧈 📅 Switch Window 📃 Window 👻 🔹 NX 12 - Mechatronics Concept Designer 🔄	$\square \times$
File	Home Modeling Assem	olies (Curve Analysis View Render Tools Application 3Dconnexion Find a Command 🔎 🔳	
Requ	irement 🐮 Mechanical C V	 Play Stop Simula 	Image: Spring Joint ↓	·
	lenu ▼ No Selection Filter ▼ En	tire Assen	nbly 🔽 🖞 🐈 🖫 ។ 🐍 🐩 😘 🖂 ។ 🕸 🧊 🔯 🗔 🗸 🖓 🥔 🖉 ។ 🕸) • •
¢	Physics Navigator		G assSortingPlant.prt ×	
	Name 🔺		🌣 Signal Adapter	υx
<u></u>	🖃 🚘 Basic Physics	^	Parameters	<u>^</u>
	osWorkpieceCube		M. Salast Dhusing Object (0)	<u>_</u>
-0-	osWorkpieceCylinder		* Select Physics Object (0)	Ψ
	+ M 😝 rbContainer		Parameter Name	Ŧ
\mathcal{O}''	+ M A rbConveyorLong	- 11	Add Parameter	*
	+ M A rbCvlinderHead		A Alias Object Object Type Parameter Value	×
P2=	- 🗹 🧰 rbCylinderLiner		paOsWorkpieceCube_SetActive osWorkpieceCube Object Source active true	
a.	🗉 🗹 🤿 rbWorkpieceCube			
Fø	💿 🗹 🍯 rbWorkpieceCylinder			•
	Joints and Constraints		Signals (2)	^
	Intermediate State S		A Name Data Time Input/Out Initial Value Measure Unit	•
9 _	fjConveyorLong		osWorkpieceCube SetActive bool Input false	T
●	fjConveyorShort	× *		X
<u> </u>	Details	v		*
* *	Dependencies	V	3 OK Apply Ca	ancel
Select	object to get its parameter		Enter a new value	

Figure 7 : Création d'un signal approprié pour un paramètre

→ Le paramètre et le signal correspondant doivent maintenant être combinés logiquement. Pour cela, faites défiler la fenêtre de commande jusqu'à la zone de commande "Formulas" (Formules). Vous voyez que vous pouvez affecter une formule au paramètre "paOsWorkpieceCube_SetActive". Cliquez pour cela sur la ligne correspondante dans le tableau (voir Figure 8, étape 1). Vous pouvez maintenant effectuer une affectation adéquate dans le champ de saisie "Formula" (Formule). Dans ce cas, une simple affectation du signal "osWorkpieceCube_SetActive" au paramètre suffit, comme représenté à la Figure 8, étape 2. Une fois que vous avez appuyé sur la touche Entrée de votre clavier, vous voyez l'affectation que vous venez de réaliser dans la colonne "Formula" (Formule) du tableau.

X 🖬 🤊 • 🤊 🖗 🗄 📾 🗱 • 🔺	Switch Window Twindow Twind
File Home Modeling Assemblies C	urve Analysis View Render Tools Application 3Dconnexion Find a Command 🔎 🗐 🛆 😮
Requirement K Mechanical C V Simulat	Image: Angular Spring Joint ↓ Image: Angular Spring Joint
∑ Menu ▼ No Selection Filter ▼ Entire Assem	bly 🔻 🕸 🐂 🐂 • 🐍 🐃 🎭 🛄 • 🕸 📦 🛛 🗊 🖉 🥒 🖉 🖉 • 🚱 • 🚱 • 🚱 •
Physics Navigator	🍯 assSortingPlant.prt 🗙
Name 🔺	Signal Adapter
Basic Physics	Signals
- SWorkpieceCylinder	A Name Data Type Input/Out Initial Value Measure Unit 🍫
🚈 🕂 🗹 🎯 rbContainer	osWorkpieceCube_SetActive_bool Input false
.≫ 💿 🗹 🍯 rbConveyorLong	
 Image: Image: Second Sec	
+ 🗹 🎯 rbCylinderHead	< > *
- 🗹 礡 rbCylinderLiner	Formulas
🕐 🗹 🏺 rbWorkpieceCube	
Image:	Assign to Formula
Joints and Constraints	paUsWorkpieceCube_SetActive osWorkpieceCube_SetActive
► Mr fjContainer	Formula
- Mr fjConveyorLong	
	osWorkpieceCube_SetActive
Details V	Name A v
Dependencies V	OK Apply Cancel
ect object to get its parameter	Enter a new value

Figure 8 : Définition d'une formule entre signal et paramètre

 REMARQUE
 Des formules complexes dépendant de plusieurs paramètres et/ou signaux peuvent également être sélectionnées. Pour une meilleure traçabilité du jumeau numérique, il est cependant recommandé de définir des formules les plus simples possibles dans l'adaptateur de signaux. La priorité doit toujours être d'intégrer la logique au programme d'automatisation et non au jumeau numérique.

Vous avez créé vous-même le premier signal de votre jumeau numérique. Créez maintenant les signaux restants en suivant la procédure décrite au <u>Chapitre 7.1</u>, "**Section : Création et connexion de signaux avec l'adaptateur de signaux**". À cet effet, utilisez les caractéristiques suivantes :

- → Le paramètre "active" (actif) avec l'alias "paOsWorkpieceCylinder_SetActive" doit être créé dans l'adaptateur de signal à partir de la source d'objet "osWorkpieceCylinder". Activez de nouveau la case à cocher en regard du paramètre pour pouvoir affecter un signal. Le signal correspondant doit avoir pour nom "osWorkpieceCylinder_SetActive" et avoir le type de données "bool". Le signal doit être défini comme "Input" (Entrée) et avoir "false" comme valeur initiale. Pour la formule du paramètre "paOsWorkpieceCylinder_SetActive".
- → Pour le paramètre "triggered" (déclenché) du capteur de collision "csLightSensorCube", créez un nouveau paramètre avec l'alias "paCsLightSensorCube_Detected" dans l'adaptateur de signaux. Pour le signal correspondant, spécifiez un signal booléen nommé "csLightSensorCube_Detected". Il doit être défini comme "Output" (Sortie) car le paramètre "paCsLightSensorCube_Detected" a la valeur "R" dans la colonne "Read/ Write" (Lecture/écriture). Affectez la valeur initiale "false" au signal. Activez la case à cocher I en regard du signal, comme montré à la Figure 9, étape 1, pour qu'une formule puisse être affectée au signal de sortie. Utilisez l'affectation directe du paramètre "paCsLight SensorCube_Detected".

NX	🖬 🤊 • 🤄 🖈 🖻 🕀 👯 •	net Switch Window 🚺 Window	w + ∓ NX 12 - Me	chatronics Concept Designer	_ 🗆 ×
File	Home Modeling Assemblies	Curve Analysis View Rende	er Tools Application 3	Dconnexion Find a Comma	ind 🔎 🗐 🛆 🕜
Requ	irement 👸 V Mechanical C V Simula	Image: Second secon	ring Joint → 👌 → ng Joint → → → nit Joint → 🐙 → cal → Elect	Operation Add	► 10, -
1 3 E	Ienu • No Selection Filter • Entire Asser	mbly 🔻 🕄 🐂 🕂 🗍		🔄 O 🍠 🤤 🖽 • 🗞	* 🗊 * 🕪 * 🛛 +
¢	Physics Navigator	assSortingPlant.prt ×			2013 S. 10
	Name 🔺	Signal Adapter			υx
9 _	+ 🗹 🎯 rbWorkpieceCylinder 🛛 🔨	A Alias	Object	Object Type Parameter	Value X ^
FE	Joints and Constraints	paCsLightSensorCube_Dete	ected csLightSensorCube	Collision Sensor triggered	false ^
2					v u
F -	→ → →	<			> +
Ø"		Signals			•
	₩ M K fjCylinderLiner	Signals			~
P1= P2=	M 😪 sjCylinderHead_CylinderLine	A Name	Data Type Input/Ou	ut Initial Value Measure	Unit 🔖
	- Carlandaria	osWorkpieceCube_SetActiv	e bool Input	false	×
B-	Couplers	osWorkpieceCylinder_SetA	ctive bool Input	false	
FO	- Sensors and Actuators	csLightSensorCube_Detect	ed bool Output	false	
M	CsLightSensorCube	<			> +
	CsLightSensorCylinder	Formulas			^
0_	sclightSensorWorknisse				
Fo	< > >	Assign to	Formula		*
\$	B 1.1	csLightSensorCube_Detected	paCsLightSensorCube_Detec	ted	, ° 🗙 🗸
÷	Details			0%	
Ŧ	Dependencies V				Cancel
Select	object to get its parameter	Enter a new	value		

Figure 9 : Création d'un signal de sortie pour un capteur photoélectrique

→ Le système de capteurs photoélectriques suivant, situé au centre du processus de transport, est constitué de deux capteurs de collision. Deux paramètres doivent donc être créés pour ce système. Tout d'abord, définissez pour le paramètre "triggered" (Déclenché) du capteur de collision "csLightSensorCylinder" un paramètre avec l'alias "paCsLightSensorCylinder_Detected" dans l'adaptateur de signaux. Créez pour le paramètre avec l'alias "paCsLightSensor CylinderTop_ Detected" dans l'adaptateur de signaux. Créez pour le paramètre avec l'alias "paCsLightSensor CylinderTop_ Detected" dans l'adaptateur de signaux. Un signal combiné réagissant aux deux paramètres peut maintenant être créé. Ce signal doit avoir pour nom "csLightSensor Cylinder_Detected" et avoir le type de données "bool". Les deux paramètres précédemment cités devant être lus également dans ce cas, le signal combiné doit être configuré comme "Output" (Sortie) et avoir pour valeur initiale "false". Spécifiez la formule suivante pour "csLightSensorCylinder_Leted", comme indiqué dans la Figure 10, étape 1 :

"((paCsLightSensorCylinderDetected) & (!paCsLightSensorCylinderTop_Detected))".

Cette formule représente une opération logique ET des deux paramètres, le second paramètre étant inversé, c'est-à-dire que le signal de sortie "csLightSensorCylinder_Detected" devient "true" lorsque "paCsLightSensorCylinderDetected" prend la valeur "true" et que "paCsLight SensorCylinderTop_Detected" prend simultanément la valeur "false". Du fait des hauteurs différentes des pièces à usiner, une pièce cylindrique est détectée uniquement lorsque le capteur inférieur de ce système de capteurs photoélectriques est déclenché et que le capteur supérieur ne détecte pas de collision en même temps (se référer aux explications sur le fonctionnement de l'installation de tri dans le Module 1 de ce cycle de formation). Cette logique est traduite par cette formule.



Figure 10 : Formule pour le signal du système de capteurs photoélectriques "csLightSensorCylinder"

REMARQUE

Lorsqu'un signal d'entrée doit être affecté à un paramètre accessible en écriture, la case à cocher en regard du paramètre 🗹 doit être activée. De la même manière, la case à cocher en regard du signal 🗹 doit être activée si un paramètre accessible en lecture doit être affecté à un signal de sortie.

Vous ne pouvez définir la formule correspondante que si une coche 🗹 est visible.

- → Pour le paramètre "triggered" (déclenché) du capteur de collision "csLightSensorWorkpiece", vous avez besoin d'un paramètre qui doit avoir l'alias "paCsLightSensor Workpiece_Detected". Pour cela, créez un signal booléen nommé "csLightSensorCube_Detected". Ce signal est également défini comme signal de sortie avec comme valeur initiale "false". La formule pour le signal "csLightSensor Workpiece_Detected" est "paCsLightSensorWorkpiece_Detected".
- → Pour le paramètre "triggered" (déclenché) du capteur de collision "csLimitSwitch CylinderNotExtended", définissez un paramètre avec l'alias "paCsLimitSwitchCylinder NotExtended_Activated" dans l'adaptateur de signaux. Ajoutez un signal booléen nommé "csLimitSwitchCylinderNot Extended_Activated". Ce signal doit être défini comme signal de sortie et avoir "false" comme valeur initiale. Spécifiez comme formule pour le signal "csLimitSwitchCylinderNotExtended_Activated" l'affectation "paCsLimitSwitchCylinder NotExtended_Activated".
- → Le paramètre "triggered" (déclenché) du dernier capteur de collision restant "csLimitSwitch CylinderRetracted" doit également être créé comme paramètre dans l'adaptateur de signaux avec l'alias "paCsLimitSwitchCylinderRetracted_Activated". Le signal correspondant "csLimitSwitch CylinderRetracted_Activated" doit avoir le type de données "bool". Il doit également être défini comme signal de sortie avec "false" comme valeur initiale. La formule pour "csLimitSwitchCylinderRetracted_Activated" doit être spécifiée comme simple affectation du paramètre "paCsLimitSwitchCylinderRetracted_Activated".
- → Pour le paramètre "active" (actif) du régulateur de position "pcCylinderHeadExtend", un paramètre avec l'alias "paPcCylinderHeadExtend_SetActive" doit être créé dans l'adaptateur de signaux. Créez ensuite un nouveau signal appelé "pcCylinderHead Extend_SetActive" de type de données "bool". Définissez le signal comme "Input" (Entrée) avec "false" comme valeur initiale. Spécifiez comme formule pour le paramètre "paPcCylinderHeadExtend_SetActive" l'affectation du signal "pcCylinderHead Extend_SetActive".

- → Pour le paramètre "active" (actif) du régulateur de position "pcCylinderHeadRetract", vous devez ensuite générer un paramètre avec l'alias "paPcCylinderHeadRetract_SetActive".
 Le signal correspondant doit avoir pour nom "pcCylinderHeadRetract_SetActive".
 Définissez ce signal comme booléen et comme "Input" (Entrée) avec "false" comme valeur initiale. La formule pour "paPcCylinderHead Retract_SetActive" est "pcCylinderHeadRetract_SetActive".
- → Le paramètre "active" (actif) du régulateur de vitesse "scConveyorLongConstSpeed" requiert un paramètre dans l'adaptateur de signaux avec l'alias "paScConveyorLong ConstSpeed_SetActive". Le signal correspondant doit avoir pour nom "scConveyorLongConst Speed_SetActive". Il doit avoir le type de données "bool", être défini comme signal de sortie et avoir "false" comme valeur initiale. L'affectation simple de "scConveyorLongConstSpeed_SetActive" doit être utilisée comme formule pour "paScConveyorLongConstSpeed_SetActive".
- → Pour le régulateur de vitesse "scConveyorLongVarSpeed", deux paramètres et deux signaux doivent être définis dans l'adaptateur de signaux.

Le premier de ces deux paramètres et son signal correspondant dans l'adaptateur de signaux servent à activer le régulateur de vitesse. Pour cela, créez pour le paramètre "active" (actif) du régulateur de vitesse "scConveyorLongVarSpeed" dans l'adaptateur de signaux un nouveau paramètre avec l'alias "paScConveyorLongVarSpeed_SetActive". Créez également le signal "scConveyorLongVar Speed_SetActive" et déclarez le type de données "bool". Le signal doit servir à la saisie. La valeur de sortie doit être "false". Enfin, spécifiez le signal "scConveyorLongVarSpeed_SetActive" comme formule pour "paScConveyorLongVarSpeed_SetActive".

La finalité du second signal est de permettre la définition variable de la vitesse de consigne du régulateur de vitesse. Pour cela, créez dans l'adaptateur de signaux un nouveau paramètre connecté au paramètre "**speed**" (vitesse) du régulateur de vitesse "**scConveyorLongVarSpeed**" et utilisez l'alias "**paScConveyorLongVarSpeed_SetSpeed**". Le signal correspondant doit avoir pour nom "**scConveyorLongVarSpeed_SetSpeed**". Comme une vitesse doit être spécifiée avec ce signal, le type de données doit être déclaré comme "**double**". Pour un signal de type non "**bool**", le type physique doit être entré sous "**Measure**" (Mesure) et l'unité physique correspondante sous "**Unit**" (Unité). Dans ce cas, assurez-vous de spécifier la valeur "**Velocity**" (Vitesse) dans la colonne "**Measure**" (Mesure) et l'expression "**mm/s**" dans la colonne "**Unit**" (Unité) pour le signal actuel (voir la <u>Figure 11</u>, étape 1). Il s'agit également d'un signal d'entrée (**Input**). Spécifiez aussi "**0.0**" comme valeur initiale. Spécifiez comme formule pour le paramètre "**paScConveyorLongVarSpeed_SetSpeed**".



Figure 11 : Création d'un signal de vitesse de type de données "double"

- → Pour le régulateur de vitesse "scConveyorShortConstSpeed", le paramètre "active" (actif) doit être inséré comme nouveau paramètre dans l'adaptateur de signaux. Donnez à ce paramètre l'alias "paScConveyorShortConstSpeed_SetActive". Créez le signal correspondant "scConveyor ShortConstSpeed_SetActive" de type de données "bool". Il doit être défini comme signal d'entrée (Input) et avoir "false" comme valeur initiale. Enfin, utilisez une affectation directe du signal "scConveyorShortConstSpeed_SetActive".
- → Le régulateur de vitesse "scConveyorShortVarSpeed" requiert également deux signaux dans l'adaptateur de signaux.

Le premier paramètre dans l'adaptateur de signaux doit se rapporter au paramètre "active" du régulateur de vitesse "scConveyorShortVarSpeed" et avoir l'alias "paScConveyor ShortVarSpeed_SetActive". Un signal booléen nommé "scConveyorShortVarSpeed_SetActive" doit être créé comme signal correspondant. Il doit être un signal d'entrée (Input) et avoir "false" comme valeur initiale. Le signal "scConveyorShortVarSpeed_SetActive" doit être affecté au paramètre "paScConveyorShortVarSpeed_SetActive".

Le second signal doit être utilisé pour définir la vitesse de consigne du régulateur de vitesse. Pour cela, créez un nouveau paramètre dans l'adaptateur de signaux basé sur le paramètre "speed" (vitesse) du régulateur de vitesse "scConveyorShortVarSpeed". Donnez à ce paramètre l'alias "paScConveyorShortVarSpeed_SetSpeed". Lorsque vous définissez le nouveau signal "scConveyorShortVarSpeed_SetSpeed", assurez-vous à nouveau de spécifier "double" comme type de données, "Velocity" (Vitesse) dans la colonne "Measure" (Mesure) et "mm/s" dans la colonne "Unit" (Unité). Il s'agit d'un signal d'entrée (Input) avec "0.0" comme valeur initiale. La formule pour le paramètre "paScConveyor ShortVarSpeed_SetSpeed". → Un autre paramètre doit être ajouté dans l'adaptateur de signaux pour garantir que la surface de transport "tsConveyorLong" se déplace uniquement lorsqu'un signal de l'un des deux régulateurs de vitesse associés est actif. Sélectionnez le nom de paramètre "active" (actif) pour cette surface de transport et donnez au nouveau paramètre l'alias "paTsConveyorLong_SetActive". Affectez ensuite la formule suivante à ce paramètre, comme décrit à la Figure 12, étape 1 :

"((scConveyorLongConstSpeed_SetActive) | (scConveyorLongVarSpeed_SetActive))"

Dans ce contexte, le caractère "|" désigne l'opération logique **OU**. Cela permet de garantir que la bande transporteuse ne se déplace que lorsqu'au moins un régulateur de vitesse a été activé pour cette dernière. La logique dans le programme d'automatisation développé prévoit cependant que la surface de transport peut uniquement être commandée par un régulateur de vitesse à la fois pendant un fonctionnement correct.



Figure 12 : Création d'un paramètre pour une surface de transport

→ Le comportement décrit précédemment pour "tsConveyorLong" s'applique également à la surface de transport "tsConveyorShort". Sélectionnez l'état "active" (actif) pour le nouveau paramètre nommé "paTsConveyorShort_SetActive".

Utilisez la formule

"((scConveyorShortConstSpeed_SetActive)|(scConveyorShortVarSpeed_SetActive))" pour garantir le bon fonctionnement de la surface de transport. Tous les paramètres et signaux nécessaires sont maintenant définis dans l'adaptateur de signaux. Affectez ensuite le nom "**saSortingPlant**" à l'adaptateur de signaux (voir <u>Figure 13</u>, étape 1). Le préfixe "**sa**" signifie "**signal adapter**", la désignation anglaise pour adaptateur de signaux. Confirmez la configuration de votre nouvel adaptateur de signaux en cliquant sur le bouton "**OK**" (voir <u>Figure 13</u>, étape 2).



Figure 13 : Création de l'adaptateur de signaux "saSortingPlant"

Une nouvelle fenêtre "Add Symbols to Symbol Table" (Ajouter des mnémoniques à la table des mnémoniques) s'affiche, dans laquelle vous êtes invité à spécifier la table des mnémoniques dans laquelle les signaux de votre adaptateur de signaux doivent être ajoutés comme mnémoniques. Ici, vous disposez de la possibilité d'étendre une table des mnémoniques existante ou de créer une nouvelle table des mnémoniques. Comme vous n'avez pas encore créé de table des mnémoniques dans votre projet, cliquez sur le bouton "New Symbol Table" (Nouvelle table des mnémoniques) (voir Figure 14, étape 1).



Figure 14 : Initier la création d'une nouvelle table des mnémoniques pour l'adaptateur de signaux

La fenêtre de commande "**Symbol Table**" (Table des mnémoniques) s'affiche. Vous pouvez y définir des nouveaux mnémoniques et affecter un nom à cette table des mnémoniques. Étant donné que vous pouvez reprendre intégralement les signaux de votre adaptateur de signaux, vous n'avez pas à définir ici de nouveaux signaux. Affectez le nom "**stSortingPlant**" à la table des mnémoniques (voir <u>Figure 15</u>, étape 1), et cliquez sur le bouton "**OK**" (voir <u>Figure 15</u>, étape 2). Le préfixe "**st**" signifie "**signal table**", la désignation anglaise pour table des mnémoniques.



Figure 15 : Terminer la création d'une nouvelle table des mnémoniques pour l'adaptateur de signaux

Vous revenez maintenant à la fenêtre "Add Symbols to Symbol Table" (Ajouter des mnémoniques à la table des mnémoniques). Si ce n'est pas déjà fait, sélectionnez la table des mnémoniques "stSortingPlant" que vous venez de créer, comme montré à la Figure 16, étape 1. Terminez le processus de création en cliquant sur le bouton "OK" (voir Figure 16, étape 2).



Figure 16 : Terminer l'affectation de mnémoniques pour l'adaptateur de signaux

Vous avez inséré tous les signaux requis dans votre modèle 3D dynamique, et vous pouvez poursuivre en établissant une connexion de signal à un API virtuel. Enregistrez d'abord les

modifications apportées au modèle en cliquant sur le bouton "Save" (Enregistrer)

7.2 Création d'une connexion de signal entre un API virtuel et un jumeau numérique

Un API virtuel doit déjà être en fonctionnement pour créer une connexion de signal. Dans cette section, vous devez donc retourner dans TIA Portal et PLCSIM Advanced. Procédez de la manière suivante pour établir la connexion :

- → Décompressez l'archive fournie avec ce module dans votre système d'exploitation (voir <u>Chapitre 7</u>) et enregistrez le contenu du dossier "fullPlcBasic" dans un dossier de votre choix. Le dossier contient le programme d'automatisation utilisé dans le Module 1 et décrit dans le Module 2.
- → Ouvrez TIA Portal et décompressez le projet "150-006_DigitalTwinAtEducation_TIAP_ Basic.zap15" dans le dossier que vous venez de créer. Procédez comme indiqué au <u>Chapitre 7.1</u> du Module 1 du cycle de formation DigitalTwin@Education.
- → Compilez la configuration matérielle et le logiciel du programme d'automatisation. Suivez pour cela les explications figurant au <u>Chapitre 7.2</u> du Module 1 de ce cycle de formation.
- → Ouvrez le programme "S7-PLCSIM Advanced" et démarrez une nouvelle instance d'un API virtuel. Donnez à cette instance le nom "DigTwinAtEdu_PLCSIM". Chargez ensuite votre programme d'automatisation dans l'API virtuel et attendez jusqu'à ce que l'état de la CPU passe à "Start", c'est-à-dire qu'une case verte s'affiche en regard du nom de l'instance. Procédez comme décrit au Chapitre 7.3 du Module 1 de ce cycle de formation.

L'API virtuel est maintenant prêt à fonctionner et vous pouvez configurer la connexion de signal au modèle 3D dynamique. Retournez à votre modèle 3D dynamique avec signaux dans Mechatronics Concept Designer et procédez comme suit :

→ Sélectionnez la commande "Signal Mapping" (Mappage de signal) dans la barre de menu "Automation" (Automatisation) (voir Figure 17, étape 1). La boîte de dialogue "Signal Mapping" (Mappage de signal) s'ouvre. Vous devez tout d'abord y sélectionner la source de signal externe. Pour cela, allez dans la zone "External Signal Type" (Type de signal externe) et sélectionnez "PLCSIM Adv" comme type, car vous souhaitez établir une connexion à PLCSIM Advanced (voir Figure 17, étape 2). À ce stade, votre modèle dynamique ne connait pas l'instance dans PLCSIM Advanced à laquelle une connexion doit être établie. Cliquez par conséquent sur le bouton "Settings" (Paramètres) sous la commande "PLCSIM Adv Instances" (Instances PLCSIM Adv) (voir Figure 17, étape 3).

NX 🖬 🔊 • 🤊 🖗 🗄 🗄 🏂 • 🗸	🔊 📅 Switch Window 🔲 Window 👻 🔹 NX 12 - Mechatronics Concept Designer 💦 _ 🗆 X
File Home Modeling Assemblies C	urve Analysis View Render Tools Application 3Dconnexion Find a Command 🔎 🖻 🐟 😮
Requirement Stop Systems Engineeri Mechanical C Simulat	Image: Spring Joint Image: Sp
T Menu - Within Work	:Part O 🔻 🗐 🐂 🐂 🗣 🛸 🕼 🗔 + 🕸 📦 🛛 🐼 ன 🖓 🖉 🖉 🖉 🖉 👘 + 🏀 + 🍘 + 🆗 +
Physics Navigator	💐 assSortingPlant.prt 🖻 🛪
Name 🔺	Signal Mapping U ×
Image: Second Systems Image: Second Systems Image: Second Systems Image: Second Systems <td>External Signal Type Type PLCSIM Adv PLCSIM Adv PLCSIM Adv Signals Signals MCD Signals (15) Find Match Case Match Whole Word Find Match Case Match Whole Word Name Adapter Name IO Type Dat osWorkpieceCu saSortingPlant Input t c</td>	External Signal Type Type PLCSIM Adv PLCSIM Adv PLCSIM Adv Signals Signals MCD Signals (15) Find Match Case Match Whole Word Find Match Case Match Whole Word Name Adapter Name IO Type Dat osWorkpieceCu saSortingPlant Input t c
scConveyorShortVarSpeed	Image: Application of the second
□ SconveyorLong	Do Auto Mapping
File Home Modeling Assemblies Curve Analysis View Render Tools Application 3Dconnexion Find a Command @ @ A Requirement Stetch Stop Angular Spring Joint Image: Stop Angular Stop Angular Spring Joint Image: Stop Angular	
_ Dependencies V	OK Cancel
•	

Figure 17 : Sélection d'une affectation de signal via PLCSIM Advanced

REMARQUE

Si des connexions de signal ont déjà été créées pour le modèle dynamique, la barre de sélection de la commande "PLCSIM Adv Instances" (Instances PLCSIM Adv) affiche toutes les instances PLCSIM Advanced qui ont été utilisées précédemment pour ce modèle. Cependant, notez que ces instances n'existent pas nécessairement et ne sont pas forcément valides. À des fins de vérification, cliquez sur le bouton "Settings" (Paramètres) et regardez l'état actuel de l'instance respective. → La fenêtre "External Signal Configuration" (Configuration de signal externe) s'ouvre. Vous pouvez y sélectionner l'instance souhaitée et valider les variables associées pour le mappage de signal. Cliquez ensuite sur le bouton "Refresh Registered Instances" (Actualiser les instances enregistrées) (voir Figure 18, étape 1). L'instance d'API virtuel que vous avez précédemment démarrée et chargée s'affiche. L'état "Run" indique que cet API virtuel est accessible. Après la sélection de l'instance comme indiqué à la Figure 18 étape 2, les signaux d'E/S pour le programme d'automatisation sont affichés. Insérez toutes les variables disponibles en sélectionnant la case à cocher "Select AII" (Tout sélectionner) (voir Figure 18, étape 3). Confirmez votre sélection en cliquant sur "OK" (voir Figure 18, étape 4).

		(1)				
stances		E			~	
		Refresh R	egistered Instance	5		
Name	ID	CPU	Status	Owner Part	Mes	
igTwinAtEdu_PLC	0	1516Fv2	Run	assSortingP	lant	
	\frown					
(2)					
	\smile					
					>	
Tags (15)						
Show		3		10	-	
Find			Match	Case 🗌 Match Who	ole Word 💊	
Select All	10	Type	Data Type	Area Type	Message	
S. Name		put	bool	Input	A	
S. Name CsLightSensorC	ube In					
S. Name csLightSensorC csLightSensorC	ube In ylin In	put	bool	Input		
S. Name csLightSensorC csLightSensorC csLightSensorW	ylin In Vork In	put put	bool bool	Input Input		
S. Name CsLightSensorC CsLightSensorC CsLightSensorW CsLimitSwitchC	ylin In Vork In Vork In	put put put	bool bool bool	Input Input Input		

Figure 18 : Valider des variables de l'instance PLCSIM Advanced pour l'affectation de signaux

 Lorsque des modifications ou des extensions sont réalisées dans le programme d'automatisation, l'instance enregistrée pour l'affectation de signal doit être mise à jour et, le cas échéant, des signaux

 REMARQUE

→ Vous revenez à la fenêtre de commande "Signal Mapping" (Mappage de signal). Vous y trouverez l'API virtuel que vous venez de sélectionner et les signaux externes disponibles dans la partie droite de la fenêtre. Vous pouvez démarrer le mappage des signaux. Sélectionnez d'abord le signal "osWorkpieceCube_SetActive" dans le tableau "MCD Signals" (Signaux MCD) à gauche dans la fenêtre (voir Figure 19, étape 1). Recherchez ensuite le signal correspondant dans le programme d'automatisation dans le tableau "External Signals" (Signaux externes). Des noms identiques ont été judicieusement choisis pour les deux programmes, comme montré à la Figure 19, étape 2. Cliquez ensuite sur le bouton "Map Signal" (Mapper le signal) pour établir une connexion entre les deux signaux (voir Figure 19, étape 3). Il est important de noter que les signaux d'entrée dans MCD ne peuvent être connectés qu'à des signaux de sortie d'un API, et vice versa.

Signal Mapping										ა
External Signal Type										
Туре								PLCSIM Ad	v	
PLCSIMAdv Instances								DigTwinAtEdu_P	LCSIN -	4
Signals	\frown									
MCD Signals (15)			^		External Signals (15)		2			
External Signal Type Type PLCSIMAdv Instances PLCSIMA PLCSIMAdv Instances PLCSIMA PLCS	ole Word									
Name	Check for N->1 Mapping									
osWorkpieceCube_SetActive	saSortingPlant	Input		0-0	csLimitSwitchCylinderRetrac	ted_Activated	Input	bool	0	^
osWorkpieceCylinder_SetActive	saSortingPlant	Input			osWorkpieceCylinder_SetAct	ive	Output	bool	0	
csLightSensorCube_Detected	saSortingPlant	Output		1.7	osWorkpieceCube_SetActive	×	Output	bool	0	
csLightSensorCylinder_Detected	saSortingPlant	Output	tput osWorkpieceCylind tput put pcCylinderHeadRet	pcCylinderHeadRetract_SetA	ctive	Output	bool	0		
<		<u></u>	>			e	<u></u>		>	Ť
			Do A	uto N	J Aapping					
1apped Signals					(3)					
Connection Name	MCD Signal Name	Direction	Exter	mal Si	ignal Name Owner Componer	nt Message				-
<									>	
		С	heck fo	or N->	1 Mapping					
								ОК	Can	cel

Figure 19 : Affectation d'un signal MCD à un signal externe

→ Vous pouvez maintenant voir le mappage de signal que vous venez de réaliser dans le tableau sous "Mapped Signals" (Signaux mappés). Insérez maintenant les autres mappages. Comme les noms des signaux MCD correspondent aux noms de variable du programme d'automatisation dans ce modèle, vous pouvez sélectionner le bouton "Do Auto Mapping" (Exécuter un mappage automatique) pour que le programme exécute ce processus automatiquement (voir Figure 20, étape 1).

Signal Mapping											υ×
External Signal Type											^
Туре									PLCSIM Adv	,	•
PLCSIMAdv Instances								[DigTwinAtEdu_PL	CSIN 🔻	1
Signals											^
MCD Signals (15)			^		External Signa	ls (15)					^
Find Match Case Match Whole Word				Find		Match Case	se 🗌 Match Whole Word 🔶				
Name	Adapter Name	IO Type	Da		Name			IO Type	Data Type	Ma	ppir
osWorkpieceCube_SetActive	saSortingPlant	Input	^	0-0	csLimitSwitch	CylinderRetrac	ted_Activated	Input	bool	0	^
osWorkpieceCylinder_SetActive	saSortingPlant	Input		1	osWorkpieceCylinder_SetActive		Output	bool	0		
csLightSensorCube_Detected	saSortingPlant	Output			osWorkpieceCube_SetActive		Output	bool	1		
csLightSensorCylinder_Detected	saSortingPlant	Output			pcCylinderHeadRetract_SetActive			Output	bool	0	
<	6 / N 1	<u></u>	>		<	15 1 C 14	- 14 	<u>.</u>		^>	•
			🗩 Do A	uto N	/lapping						
Mapped Signals	(1	\rangle									^
Connection Name MCI		MCD Signal Name		Name	Direction	External Si	External Signal Name		ient /	\$	
PLCSIM Adv.DigTwinAtEdu_PLCSIM											
✓ saSortingPlant_osWorkpieceCube_SetActive_osWorkpieceCube_SetActive osWorkpieceCube_SetActive			piece	Cube_SetActive	←	osWorkpie	ceCube_SetActive				
1											
			Chack fo	r NLS	1 Manning						_
			CHECKIC	14-2	r mapping						
									OK	Can	cel
									OK	Call	cer

Figure 20 : Connecter tous les signaux par mappage automatique



Support d'apprentissage/de formation | Module DigitalTwin@Education 150-006 | Édition 07/2020 | Digital Industries, FA

Mapped Signals		(1)						
Connection Name	MCD Signal Name	Direction	External Signal Name					
🖃 🗹 PLCSIM Adv.DigTwinAtEdu_PLCSIM								
saSortingPlant_osWorkpieceCube	osWorkpieceCube	←	osWorkpieceCube_Se					
Check for N->1 Mapping								

Figure 21 : Couper le mappage du signal

→ Le mappage automatique a maintenant interconnecté les 15 signaux entre le modèle 3D dynamique et l'API virtuel. Vérifiez si les mappages sont corrects et terminez le mappage de signal en cliquant sur le bouton "OK" (voir Figure 22, étape 1).

Signal Mapping											ა x
External Signal Type											^
Туре									PLCSIM Adv		•
PLCSIMAdv Instances									DigTwinAtEdu_PLCS	IN 🕶	Þ
Signals											^
MCD Signals (15)			^		External Signa	ls (15)					^
Find Match Case Match Whole Word			rd 🔶		Find		Match Cas		e 🗌 Match Whole Word 🗼		
Name	Adapter Name	Ю Туре	Da		Name			Ю Туре	Data Type	Ma	ppir
osWorkpieceCube_SetActive	saSortingPlant	Input	^	848 1	csLimitSwitch	nCylinderNotExt	ended_Activated	Input	bool	1	^
osWorkpieceCylinder_SetActive	saSortingPlant	Input			csLimitSwitchCylinderRetracted_Activated		Input	bool	1	10	
csLightSensorCube_Detected	saSortingPlant	Output			osWorkpieceCylinder_SetActive		Output	bool	1		
csLightSensorCylinder_Detected	saSortingPlant	Output			osWorkpiece	eceCube_SetActive		Output	bool	1	
<		· · ·	>		<			^ · · ·		ं	Ť
			Do A	uto N	Napping						
Mapped Signals											~
Connection Name			MCD Si	gnal N	Name	Direction	External Sigr	nal Name	Owner Componen	t M	\$
- V PLCSIM Adv.DigTwinAtEdu_PLCSIM										^	-
saSortingPlant_osWorkpieceCube_SetActive_osWorkpieceCube_SetActive_os			osWorkpieceCube_SetActive ←			←	osWorkpieceCube_SetActive				
saSortingPlant_osWorkpieceCylinder_SetActive_osWorkpieceCylinder_Set			osWorkpieceCylinder_SetAc osWorkpieceCy				Cylinder_SetAc	ider_SetAc			
<									1	>	
			Check fo	or N->	1 Mapping						
							(1)		Can	cel

Figure 22 : Confirmation du signal entre modèle dynamique et API virtuel

La connexion entre le modèle 3D dynamique dans NX/MCD et le programme d'automatisation dans votre API virtuel est maintenant établie. Enregistrez votre modèle en cliquant sur le bouton

"Save" (Enregistrer)

7.3 Test du jumeau numérique avec l'API virtuel

Dans ce chapitre, vous devez exploiter votre jumeau numérique en interaction avec un programme d'automatisation dans un API virtuel et valider la fonctionnalité. Pour cela, procédez comme suit :

- → Après avoir chargé le programme d'automatisation dans une instance d'un API virtuel au <u>Chapitre 7.2</u>, vous pouvez maintenant démarrer l'IHM à l'aide de l'outil de simulation "WinCC Runtime Advanced". Pour ce faire, utilisez TIA Portal Utilisez la procédure décrite au <u>Chapitre 7.4</u> du Module 1 du cycle de formation DigitalTwin@ Education.
- → Allez ensuite dans le programme "Mechatronics Concept Designer" et lancez une simulation pour votre jumeau numérique. Exécutez la commande "Play" (Démarrer) dans la barre de commande "Simulation".
- → Exécutez les deux scénarios de test du premier module de ce cycle de formation pour votre jumeau numérique et validez la fonctionnalité dans votre jumeau numérique. Suivez les descriptions du <u>Chapitre 7.6</u> du Module 1 de ce cycle de formation. Vous pouvez voir que le jumeau numérique que vous avez créé vous-même dans les Modules 4 6 de ce cycle de formation se comporte de la même manière que le modèle spécifié que vous avez utilisé dans les trois premiers modules. À la fin de votre série de tests, arrêtez la simulation dans MCD, quittez l'instance d'IHM simulée et fermez votre API virtuel.

Bien entendu, vous pouvez vérifier votre jumeau numérique avec votre programme d'automatisation optimisé du **Module 3**.

Vous êtes arrivé à la fin de ce module de formation. Avec les connaissances acquises, vous pouvez créer vous-même votre propre jumeau numérique et exécuter une mise en service virtuelle pour votre projet d'automatisation.

8 Liste de contrôle – Instructions structurées par étapes

La liste de contrôle suivante permet aux stagiaires/participants à la formation de vérifier euxmêmes si toutes les étapes de travail des instructions ont été minutieusement réalisées et permet de clore le module avec succès de façon autonome.

N٥	Description	Vérifié
1	Votre modèle dynamique du Module 5 a été étendu pour inclure les signaux requis.	
2	Une connexion de signal valide a été créée entre votre jumeau numérique et l'API virtuel.	
3	Le jumeau numérique que vous avez créé vous-même a été entièrement validé via une simulation des scénarios de test du Module 1 de ce cycle de formation.	

Tableau 1 : Liste de contrôle de la "Création de signaux pour un modèle 3D dynamique dans le système d'IAO Mechatronics Concept Designer"

9 Informations supplémentaires

Des informations complémentaires vous sont proposées afin de vous aider à vous exercer ou à titre d'approfondissement, par ex. : mises en route, vidéos, didacticiels, applis, manuels, guides de programmation et logiciel/firmware d'évaluation sous les liens suivants :

Aperçu "Informations complémentaires" - En préparation

Vous trouverez ci-après des liens intéressants :

- [1] <u>support.industry.siemens.com/cs/document/90885040/programming-guideline-for-s7-1200-s7-1500?dti=0&lc=en-US</u>
- [2] <u>support.industry.siemens.com/cs/document/109756737/guide-to-standardization?dti=0&lc=en-US</u>
- [3] omg.org/spec/UML/2.5.1/PDF
- [4] geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/
- [5] geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/

Pour plus d'informations...

Siemens Automation Cooperates with Education siemens.com/sce

Supports d'apprentissage/de formation SCE siemens.com/sce/documents

Packs pour formateurs SCE siemens.com/sce/tp

Contact partenaire SCE siemens.com/sce/contact

Digital Enterprise siemens.com/digital-enterprise

Totally Integrated Automation (TIA) siemens.com/tia

TIA Portal siemens.com/tia-portal

TIA Selection Tool siemens.com/tia/tia-selection-tool

Automate SIMATIC siemens.com/controller

Documentation technique SIMATIC siemens.com/simatic-docu

Industry Online Support support.industry.siemens.com

Système de catalogue et de commande Industry Mall mall.industry.siemens.com

Siemens Digital Industries, FA Postfach 4848 D-90026 Nürnberg Allemagne

Sous réserve de modifications et d'erreurs © Siemens 2020

siemens.com/sce